

## **CLAIMS**

I claim:

- 5 1. A method of creating a grammar for a natural language dialog system from a domain model, said method comprising the steps of:
  - a) creating instances of general purpose grammar rules as a new grammar, said general purpose grammar rules including a plurality of selected objects, each of said objects including one or more attributes;
  - 10 b) creating an umbrella rule for each broad category of queries in said general purpose grammar rules;
  - c) selectively including objects as domain objects in said new grammar;
  - d) creating umbrella rules for domain object attributes; and
  - e) selectively including attributes in said new grammar.
- 15 2. A method as claim 1, wherein the step (a) of creating the new grammar comprises the steps of:
  - i) creating an initially empty new grammar;
  - 20 ii) opening a template grammar, said template grammar including parameterized general purpose rules to be instantiated;
  - iii) creating instances of said general purpose grammar rules; and
  - iv) adding the instantiated general purpose rules to said new grammar.
- 25 3. A method as in claim 1, wherein each said umbrella rule includes a domain-object-independent non-terminal in a left-hand side and a set of expansions of said non-terminal in a right-hand side.
4. A method as in claim 3, wherein each of said set of non-terminal expansions is  
30 a domain-specific instantiation of the broad category.
5. A method as in claim 3, wherein the domain-object-independent non-terminal relates the umbrella rule to a broad category of rules.

6. A method as in claim 1, wherein the step (c) of selectively including objects comprises the steps of:

- i) selecting an object from said plurality of objects; and
- ii) presenting said selected object to a developer, said developer deciding whether said object is included in said new grammar.

7. A method as in claim 6, wherein if said developer decides that said selected object should be included in said grammar, said method further comprises the steps of:

- c1) creating an umbrella rule for each broad category of object phrases;
- c2) creating an object name grammar rule for each created umbrella rule;
- c3) allowing the developer to select names for the included object; and
- c4) adding entries in said grammar for each selected name.

8. A method as in claim 7, wherein the object name grammar rules created in step (c2) further includes an object name possessive rule, said object name possessive rule specifying the possessive form of the name of said object.

9. A method as in claim 8, said method further comprising the steps of:

- c5) specializing object rules not requiring developer input; and
- c6) specializing query rules requiring only the name of the domain object.

10. A method as in claim 1, wherein in step (e) the developer selects whether an attribute is included in the new grammar.

11. A method as in claim 10, wherein if the developer selects an attribute for inclusion, said method further comprises the steps of:

- f) including attribute name grammar rules naming said attributes;
- g) identifying whether said named attribute is complex or simple.

12 A method as in claim 11, wherein in step (g) if an attribute is identified as complex, said method comprising the steps of:

- vi) creating grammar rules relating the object including said complex attribute
- 5 to a subsidiary domain object; and
- vii) adding said created relating grammar rules to said new grammar.

13. A method as in claim 11, wherein in step (g) if said attribute is identified as simple, said method further comprising the steps of:

- 10 vi) creating grammar rules for the atomic values of the simple attribute;
- vii) adding said created atomic value grammar rules to said new grammar;
- viii) creating grammar rules requiring the name of the object including the simple attribute and the atomic value of the domain attribute; and
- ix) adding said grammar rules created in step (iii) to said new grammar.

15

14. A method as in claim 1, said method further comprising the step of:

- (h) detecting and repairing inconsistencies in the newly-created grammar.

15. A method as in claim 14, wherein the step (f) of detecting and repairing inconsistencies comprises the steps of:

- 20 i) running a grammar checker on the new grammar, said grammar checker identifying unreachable non-terminals;
- ii) repairing said newly-created grammar to eliminate identified unreachable non-terminals;
- 25 iii) running said grammar checker on said repaired new grammar, said grammar checker identifying non-terminating expansions; and
- iv) repairing said new grammar to eliminate identified non-terminating expansions.

30

16. A method as in claim 15 wherein the step (ii) of eliminating unreachable non-terminals comprises the steps of:

- F) prompting the developer to delete unreachable non-terminals; and
- G) prompting the developer to add new rules to make all remaining
- 35 non-terminals reachable.

17. A method as in claim 15, wherein the step (ii) of eliminating non-terminating expansions comprises the steps of:

F) prompting the developer to delete identified non-terminating expansions;

and

G) prompting the developer to add rules terminating all remaining said non-terminating expansions.

18. A system for creating a natural language dialog grammar from a domain model, said system comprising:

means for forming a new grammar including instances of general purpose grammar rules, said instances of general purpose rules being a plurality of objects, each of said objects including one or more attributes;

means for creating umbrella rules for query categories and attributes;  
means for selecting objects included as domain objects in said new grammar;  
means for selecting attributes included in said new grammar; and  
means for detecting and repairing inconsistencies in said new grammar.

19. A system as in claim 18, wherein the means for forming the new grammar comprises:

means for creating an initially empty new grammar;

means for opening a template grammar including parameterized general purpose rules;

means for creating instances of general purpose rules; and  
means for adding the general purpose rules to said new grammar.

20. A system as in claim 19, wherein said means for creating umbrella rules creates a rule with a left-hand side being a domain-object-independent non-terminal and a right-hand side including a set of expansions of said non-terminal, each of said set of non-terminal expansions being a domain-specific instantiation of the broad category, said non-terminal relating the rule to a broad category of rules.

21. A system as in claim 18, wherein the means for selectively including objects comprises:  
means for selecting objects;  
means for presenting said selected objects to a developer, said developer  
5 deciding whether said presented objects are included in said new grammar.
22. A system as in claim 21, further comprising:  
means for creating an umbrella rule for a broad category of object phrases;  
means for creating object name grammar rules for umbrella rules;  
10 means for receiving a name selection from a developer; and  
means for entering names in said grammar responsive to selection by said developer.
23. A system as in claim 22, wherein the means for creating object name rules  
15 further includes means for creating an object name possessive rule, said object name possessive rule specifying the possessive form of the name of said object.
24. A system as in claim 23, said system further comprising:  
means for automatically specializing object rules; and  
20 means for specializing query rules responsive only to the name of the domain object.
25. A system as in claim 18, wherein the means for selecting attributes included in the new grammar comprises:  
25 means for presenting said attributes to a developer, said developer deciding whether each of said presented attributes is included.
26. A system as in claim 25, said system further comprising:  
means for generating attribute name grammar rules for selected said attributes;  
30 complex attribute grammar rule creation means for identifying attributes, relating the object to a subsidiary domain object as complex attributes, grammar rules being created for said complex attributes and added to said new grammar;  
atomic value grammar rule creation means for identifying simple attributes, creating grammar rules for atomic values of identified simple attributes and, adding  
35 said atomic value grammar rules to said new grammar; and

simple attribute grammar rule creation means for creating simple attribute grammar rules requiring the name of simple objects and a corresponding atomic value and adding said simple attribute grammar rules to said new grammar.

- 5 27. A system as in claim 18, wherein the means for detecting and repairing inconsistencies comprises:
- means for identifying unreachable non-terminals;
  - means for eliminating unreachable non-terminals from said new grammar;
  - means for identifying non-terminating expansions; and
  - 10 means for eliminating non-terminating expansions from said new grammar.
28. A system as in claim 27, wherein the means for eliminating unreachable non-terminals comprises:
- means for allowing a developer to delete identified unreachable non-terminals;
  - 15 and
  - means for allowing the developer to add new rules making identified unreachable non-terminals reachable.
29. A system as in claim 28, wherein the means for eliminating non-terminating expansions comprises:
- means for allowing the developer to delete identified non-terminating expansions; and
  - means for allowing the developer to add rules terminating said identified non-terminating expansions.
  - 25
30. A computer program product for creating a natural language dialog grammar from a domain model, said computer program product comprising a computer usable medium having computer readable program code thereon, said computer readable program code comprising:
- 30 computer readable program code means for forming a new grammar including instances of general purpose grammar rules, said instances of general purpose rules being a plurality of objects, each of said objects including one or more attributes;
  - computer readable program code means for creating umbrella rules, umbrella rules being created for query categories and for attributes; and

computer readable program code means for selecting objects and attributes included as domain objects and domain object attributes in said new grammar.

31. A computer readable program code means for creating a natural language grammar from domain models as in claim 30, wherein the computer readable program code means for forming the new grammar comprises:

computer readable program code means for creating an initially empty new grammar;

computer readable program code means for opening a template grammar including parameterized general purpose rules to be instantiated;

computer readable program code means for creating instances of general purpose rules; and

computer readable program code means for adding the instantiated general purpose rules to said new grammar.

15

32. A computer readable program code means for creating a natural language grammar from domain models as in claim 31, wherein said computer readable program code means for creating umbrella rules creates rules with a left-hand side being a domain-object-independent non-terminal and a right-hand side including a set of expansions of said non-terminal, each of said set of non-terminal expansions being a domain-specific instantiation of the broad category, said non-terminal relating the rule to a broad category of rules.

20

33. A computer readable program code means for creating a natural language grammar from domain models as in claim 32, wherein the computer readable program code means for selecting objects and attributes comprises:

25

computer readable program code means for individually selecting objects; and

computer readable program code means for presenting said selected objects to a developer, said developer deciding whether said presented objects are included in said new grammar.

30

34. A computer readable program code means for creating a natural language grammar from domain models as in claim 33, wherein the computer readable program code means for selecting objects and attributes further comprises:

computer readable program code means for presenting selected said attributes to a developer, said developer deciding whether each of said presented attributes is included in said new grammar.

5     35.     A computer readable program code means for creating a natural language grammar from domain models as in claim 34, wherein the computer readable program code means for creating umbrella rules further creates umbrella rules for a broad category of object phrases, said computer readable program code means further comprising:

10             computer readable program code means for creating object name grammar rules for umbrella rules;

              computer readable program code means for receiving a name selection from a developer, said name selection selectively including said object name created by said computer responsive to creating said object name grammar rules; and

15             computer readable program code means for entering names in said new grammar responsive to selection by said developer.

20     36.     A computer readable program code means for creating a natural language grammar from domain models as in claim 35, wherein the computer readable program code means for creating object name rules further includes computer readable program code means for creating an object name possessive rule, said object name possessive rule specifying the possessive form of the name of said object.

25     37.     A computer readable program code means for creating a natural language grammar from domain models as in claim 36, said system further comprising:

              computer readable program code means for automatically specializing object rules; and

              computer readable program code means for specializing query rules responsive only to the name of the domain object.

30

38.     A computer readable program code means for creating a natural language grammar from domain models as in claim 35, further comprising:

              computer readable program code means for generating attribute name grammar rules for selected said attributes;



computer readable program code means for identifying attributes relating to a subsidiary domain object as complex attributes, grammar rules being created for said complex attributes and added to said new grammar;

computer readable program code means for identifying simple attributes,  
5 creating grammar rules for atomic values of simple attributes, and adding said atomic value grammar rules to said new grammar; and

computer readable program code means for creating simple attribute grammar rules requiring the name of simple objects and a corresponding atomic value and, adding said simple attribute grammar rules to said new grammar.

10

39. A computer readable program code means for creating a natural language grammar from domain models as in claim 30, said system further comprising:

computer readable program code means for detecting and repairing inconsistencies in newly-created grammar.

15

40. A computer readable program code means for creating a natural language grammar from domain models as in claim 39, wherein the computer readable program code means for detecting and repairing inconsistencies comprises:

20 computer readable program code means for identifying unreachable non-terminals;

computer readable program code means for eliminating unreachable non-terminals;

computer readable program code means for identifying non-terminating expansions; and

25 computer readable program code means for eliminating non-terminating expansions.

41. A computer readable program code means for creating a natural language grammar from domain models as in claim 40, wherein the computer readable program  
30 code means for eliminating unreachable non-terminals comprises:

computer readable program code means for selectively deleting identified unreachable non-terminals; and

computer readable program code means for selectively adding new rules making identified unreachable non-terminals reachable.

35

42. A computer readable program code means for creating a natural language grammar from domain models as in claim 41, wherein the computer readable program code means for eliminating non-terminating expansions comprises:

- 5 computer readable program code means for selectively deleting identified non-terminating expansions; and
- computer readable program code means for selectively adding rules terminating identified non-terminating expansions.